

### REMARKS

The comments of the applicant below are each preceded by related comments of the examiner, shown in small, bold type.

**6. ... Claims 11-48, 75 and 89-145 are rejected under 2173.05(m) PROLIX.**

In a telephone conference between David Feigenbaum and the examiner in July 8, 2004, the Examiner indicated that the applicant need not reply to this rejection.

**8. Claims 1 - 8, 76— 87, 102— 104 are rejected under 35 U.S.C. 102(b) as being anticipated by Microsoft's commercial product Microsoft Foundation Class (MFC) library implemented in C++ as disclosed by Chapter 17 of the text book, "MFC with Visual C++ 5", by Mike Blaszcak, published April 19, 1997.**

**NOTE: This rejection is written to the level of a programmer. The MFC product is a programming library. The sections of the product selected for the rejection cover the use of proxies for collecting input from users and loading data in containers. Furthermore, the rejection covers the settings in MFC for features such as focus the ability to determine the focus on the screen and event handlers.**

#### **Claim 1**

**MFC anticipates in a computing system, a method for tracking user interaction with computer readable code, said method comprising: monitoring user interaction with computer readable code (MFC, pages 821 - 822, Event Notifications - event firing and the ability to detect such an event); recording information about the type of the user interaction (MFC, Control Proxy classes - Containers — pages 809—814) contain; if the user interaction relates to a point of focus in a first position in the computer readable code, then recording information about said first position; if the point of focus changes from said first position to a second position in the computer readable code(MFC, pages 821 - 822, Event Notifications - event firing and the ability to detect such an event), then determining if the content of the point of focus in said first position has changed(MFC, pages 821 - 822, Event Notifications - event firing and the ability to detect such an event); and if the content of the point of focus in said first position has changed(MFC, pages 821 - 822, Event Notifications - event firing and the ability to detect such an event), then recording information about the content of the point of focus in said first position (MFC, Control Proxy classes - Containers — pages 809—814).**

Claim 1 has been amended for clarity and to recite that the proxy component is generated by analyzing the first component to determine the type of that component. The MFC reference neither discloses nor suggests "analyzing the first component to determine its type." In the MFC reference, the proxy component is generated based simply on a single method of mapping user

level events in ActiveX controls to notification endpoints within a Visual C++ program, duplicating the inputs and outputs of the ActiveX control.

In a particular example described in the applicant's specification, by contrast, a component resides in a first programming domain, and is not necessarily intended to be used in any other programming domain. To generate a proxy allowing the use of that component in a different programming domain, the process analyzes specific attributes of the component, such as "the type of the first component ... For example, if the first domain is the Java programming language, then the first component may be identified as a Java class or a Java interface." (p. 68, ll. 16-18) Based on this analysis, the process selects the "appropriate transformation to be performed on the first component...[which] often depends on the type of the component." (p. 68, ll. 20-21) That transformation is applied "to transform [the] component of the first domain to a proxy component of [the] second domain." (p. 71, ll. 1-2) To assure that the concept is accurately represented, the transformation produces a proxy "that adhere[s] to the syntax of the second domain and [has] a semantic usability in the second domain closely corresponding to the semantic usability of the first component in the first domain." (p. 71, ll. 5-7) That is, the proxy has attributes in the second domain that closely correspond to the attributes of the component in the first domain. This allows full use to be made of the component, via the proxy, within the second domain, as if the component were natively available in that domain, without any requirement that that component be designed for such cross-domain use. A wide variety of other examples are within the scope of the claims.

Claims 76-79, 102-104, and the dependent claims are patentable for at least the same or similar reasons.

**10. Claims 9—48, 75 and 88— 101 and 105 - 145 rejected under 35 USC. 103(a) as being unpatentable over MFC as per above in view of Essential JNI JAVA Native Interface by Rob Gordon published 1998.**

**Claim 9**

**MFC teaches the computer program product of claim 1, wherein and the second functional domain is the C++ programming language as per claim 1, but MFC does not teach the first functional domain is the Java programming language. It is JNI who teaches the ability to communication between C++ and JAVA (v.v.) (JNI, pages 249 and page 270 and 278). Therefore it would have been obvious to one of ordinary skill in the art at the time of invention to combine MFC and JNI , because**

Applicant : Alexander R. Krapf et al.  
Serial No. : 09/551,246  
Filed : April 17, 2000  
Page : 25 of 25

Attorney's Docket No.: 12915-002001

**the ability to share container information from user input makes the language of implementation transparent to the user.**

Claim 9 and the dependent claims are patentable for at least the same or similar reasons as claim 1.

The fact that the applicant has addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner that have not been addressed. The fact that the applicant has made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims. The fact that the applicant has made claim amendments does not mean that the applicant concedes any of the examiner's positions with respect to those claims.

Enclosed is a \$475.00 check for the Petition for Extension of Time fee. Please apply any other charges or credits to deposit account 06-1050, reference 12915-002001.

Respectfully submitted,

Date: \_\_\_\_\_

7/22/14



David L. Feigenbaum  
Reg. No. 30,378

Fish & Richardson P.C.  
225 Franklin Street  
Boston, MA 02110-2804  
Telephone: (617) 542-5070  
Facsimile: (617) 542-8906